X-Authentication-Warning: teak.ii.uib.no: larsr owned process doing -bs
Date: Mon, 15 May 2000 14:38:35 +0200 (MET DST)
From: Lars Ramkilde Knudsen <Lars.Knudsen@ii.uib.no>
To: AESRound2@nist.gov
cc: Lars Ramkilde Knudsen <Lars.Knudsen@ii.uib.no>
Subject: twofish

Dear NIST,

Please find enclosed some pages on Twofish.

Lars R. Knudsen, Assoc.Prof., Univ. of Bergen, Dept.of Informatics, PB
7800, N-5020 Bergen, Norway +47 55 58 41 57, (fax +47 55 58 41 99),
Lars.Knudsen@ii.uib.no, http://www.ii.uib.no/~larsr/

Trawling Twofish (revisited)[*]

Lars R. Knudsen
Dept. of Informatics
University of Bergen, Norway

May 15, 2000

Abstract

Twofish is a 128-bit block cipher submitted as a candidate for the Advanced Encryption Standard (AES). It has a structure related to the Feistel structure and runs in 16 rounds. In this paper we consider mainly differentials of Twofish and show that there are differentials for Twofish for up to 16 rounds, predicting at least 32 bits of nontrivial information in every round. In addition, it holds that for any fixed user-selected key it is possible, at least in theory, to find one good pair of plaintexts following the differential through all 16 rounds. Also, we use these findings to try and distinguish (reduced) Twofish from a randomly chosen permutation.

1 Introduction

Twofish [16] is a secret-key encryption primitive, which is one of the final five candidates for the Advanced Encryption Standard [15]. Twofish is a 16-round cipher which uses components from the ciphers Khufu [14], Square [1], and SAFER [13]. The 128-bit plaintexts are first split into four words of each 32 bits, $X^0_{LL}, X^0_{LR}, X^0_{RL}$, and X^0_{RR}. The four words are then exclusive-or'ed with the 32-bit round keys, K_0, K_1, K_2, and K_3 respectively. Then we compute for $i = 0, \ldots, 15$:

$$\begin{aligned} w_1 &= \mathbf{g}(X^i_{LL}) && (1)\\ w_2 &= \mathbf{g}(X^i_{LR} << 8) && (2)\\ X^{i+1}_{LL} &= ((w_1 + w_2 + K_{2i+8}) \oplus X^i_{RL}) >> 1 && (3)\\ X^{i+1}_{LR} &= (w_1 + 2w_2 + K_{2i+9}) \oplus (X^i_{RR} << 1) && (4)\\ X^{i+1}_{RL} &= X^i_{LL} && (5)\\ X^{i+1}_{RR} &= X^i_{LR} && (6) \end{aligned}$$

The function $\mathbf{g}$ consists of four key-dependent S-boxes plus a linear transformation derived from an MDS-code. The key-dependent S-boxes are computed from two fixed 8-bit S-boxes q_0 and q_1. The ciphertext is the concatenation of the values $X^{16}_{RL} \oplus K_4, X^{16}_{RR} \oplus K_5, X^{16}_{LL} \oplus K_6, X^{16}_{LR} \oplus K_7$. Figure 1 is a non-detailed picture of one round of Twofish. Here $\mathbf{g}_8$ is the same as $\mathbf{g}$ but where the inputs are rotated by eight positions to the left. For a more detailed pictorial illustration of the encryption function of Twofish we refer to Figure 1 of [16].

It follows by inspection of Figure 1 that Twofish does not have the classical Feistel structure as claimed. If one removes the one-bit rotations, then the resulting cipher is a Feistel cipher. Although it is possible to incorporate the one-bit rotations inside the round function by applying simple transformations, this would result in different round functions in the different rounds.

[*] Revised version of [4].

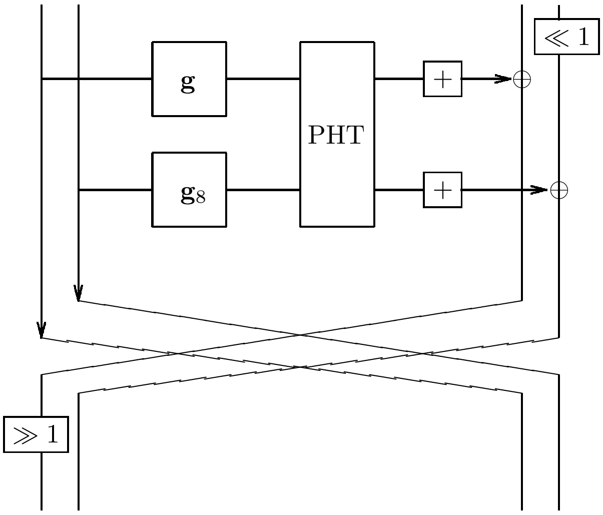

Figure 1: The Twofish graph. $\boxed{+}$ denotes the addition of a round key modulo 2^{32}, $\mathbf{g}_8$ is the function $\mathbf{g}$ but where the inputs are rotated 8 positions to the left, and $\text{PHT}(x, y) = (x + y, x + 2y)$ both outputs modulo 2^{32}.

2 The Twofish S-boxes q_0 and q_1

In this section we analyse the fixed S-boxes used in Twofish. Each of the S-boxes q_0 and q_1 are constructed from four 4-bit S-boxes, t_1, t_2, t_3, and t_4. Call this construction the $q_{0,1}$-construction. The following fact is well-known.

Fact 1 *Consider a function $f : \{0, 1\}^r \to \{0, 1\}^r$. If f is a permutation, then the algebraic degree of any output bit as a function of the input bits is at most $r - 1$.*

Now we can prove the following property of the S-boxes.

Fact 2 *For any choices of the bijective 4-bit S-boxes t_1, t_2, t_3, and t_4 in the $q_{0,1}$-construction each bit in the output of the resulting 8-bit S-box can be written as a function of the input bits with algebraic degree at most six.*

Proof. The left half of the 8-bit input in the $q_{0,1}$-construction maps one-to-one to both the left and right halves of the output, and similarly for the right half of the input. Therefore, any output bit will be of at most degree three as a function of either half of the input, totally at most degree six. □

Note that the nonlinear order of an S-box is not the same as the algebraic degree of the output bits. The nonlinear order of an n-bit bijective S-box is the minimum algebraic degree of the $2^n - 1$ boolean functions obtained from a linear combination of the n coordinate functions. It can be shown that the nonlinear order of a randomly chosen bijective n-bit S-box is $n - 2$ with a high probability.

The authors of Twofish note [16, §7.2.1] "The construction method for building q_0 and q_1 from 4-bit permutations was chosen because .... without adding any apparent weaknesses to the cipher". For a randomly chosen 8-bit permutation the probability is very high that the algebraic degree of one or more output bits as functions of the input bits is seven. So random 8-bit S-boxes have better properties than the $q_{0,1}$-construction with respect to the algebraic degrees, the question is if they are substantially better.

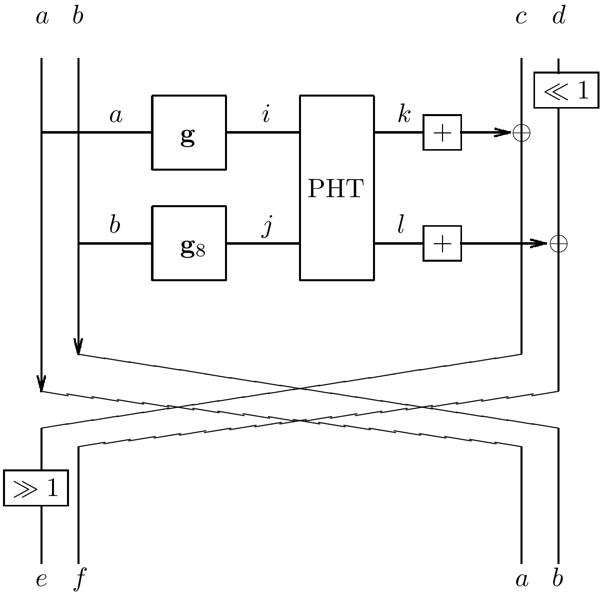

Figure 2: A one-round truncated differential.

3 The linear transformation

The linear transformation inside the function **g** of Twofish is similar to the constructions in Square [1] and Rijndael [2]. It is a permutation of a vector of four bytes, such that for any two different input vectors, the total number of different bytes in the input vector and the output vector is at least five. This provides diffusion to the cipher. Two inputs to **g** different in only one of the four bytes are guaranteed to differ in all four bytes at the output of **g**. However, this also enables an attacker to specify a so-called "truncated differential", see e.g. [6, 7, 11], of probability one through the **g** transformation. Let (x, y, z, w) denote the difference of two vectors each of four bytes for any definition of difference, and let $(x, y, z, w) \xrightarrow{g} (X, Y, Z, W)$ denote that two input vectors of differences x, y, z, and w in the four bytes can lead to outputs of differences X, Y, Z, and W in the four bytes. Then the diffusion property implies that for $a \neq 0$ the differential $(a, 0, 0, 0) \xrightarrow{g} (b_0, b_1, b_2, b_3)$ where $b_i \neq 0$ for $i = 0, \ldots 3$ holds with probability one.

4 Differentials for Twofish

In this section we consider (truncated) differentials of Twofish. The differentials will specify the expected differences in each of the 32-bit words in the intermediate ciphertexts. Let us introduce some notation. We define the difference between two 32-bit words, X and Y as

$$X - Y \bmod 2^{32}.$$

With this definition, the difference before and after the addition of a round key is the same. Also, the PHT-transform is linear with respect to this difference.

We shall write a one-round differential as

$$(a, b, c, d) \to (e, f, a, b) : (a, b) \to (i, j) \to (k, l), \tag{7}$$

where all small letters denote a difference of two 32-bit words. The first two tuples represent the differences in the four input words and in the four output words of the particular round, see Figure 2. Here $k = i + j \bmod 2^{32}$ and $l = k + j \bmod 2^{32}$. The three following pairs specify first the differences in the inputs to the **g**-functions, then the differences in the words before and after the PHT-transform, respectively. Also, at first we shall only be interested in whether the difference in a 32-bit word is zero or nonzero.

Here we give a two-round differential, Ω_1, which has probability 2^{-32} and gives non-trivial information about at least 64 bits of the (intermediate) ciphertexts.

$$\begin{array}{lcll} (a,0,c,d) \rightarrow (e,f,a,0) & : & (a,0) \rightarrow (i,0) \rightarrow (i,i), & p=1 \\ (e,f,a,0) \rightarrow (h,0,e,f) & : & (e,f) \rightarrow (2m,-m) \rightarrow (m,0), & p=2^{-32} \end{array}$$

The differential is iterative, that is, it can be concatenated with itself any number of times. If we start and end with one-round differentials of probability one, this yields $(2r+1)$-round differentials of probability $2^{-(32r)}$. Some explanation of the differential. The only requirement on the input differences is that $(a,c,d) \neq (0,0,0)$. Then $a \neq 0 \Rightarrow i \neq 0$, and $e \neq 0, f \neq 0 \Rightarrow m \neq 0$. In the first round, two inputs of nonzero difference a to $\mathbf{g}$ yields some nonzero difference i in the outputs of $\mathbf{g}$, and since the PHT is linear with respect to the difference used, it follows that the differences in both 32-bits words after addition of the round keys will be i. Note that since the outputs of the round function are combined with the right halves of the inputs to the round using the exclusive-or operation, the differences e and f are not just $c \oplus i$ and $d \oplus i$, respectively. More precisely, let e_1 and e_2 be the two texts of difference e and similarly for the other words. Then $e = e_1 - e_2 = (i_1 \oplus c_1) - (i_2 \oplus c_2)$ (here we ignored the one-bit rotations), where $i = i_1 - i_2$ and $c = c_1 - c_2$. However, if the values of c and d are nonrandom and related, then so are the values of e and f. This phenonmenon will be discussed later in this paper. In the second round, the two pairs of inputs of differences e and f, respectively, lead to differences $2m$ and $-m$ with an average probability of 2^{-32}, where we assumed that neither e nor f is zero, which will happen with probability roughly $(1-2^{-31})$ if c, d are random. Note also, that $(e,f) = (0,0)$ with probabilty 2^{-64} if c, d are random, in which case the (rest of the) second round has a probability of one. All in all, the probability of the second round is approximately 2^{-32}.

Summing up, the differential predicts 32 bits of information in each of two rounds, a total of 64 bits with a probability of (about) 2^{-32}.

For Ω_1 a pair of plaintexts will have a difference of $(a,0,c,d)$, where $(a,c,d) \neq (0,0,0)$. Thus, it is possible to generate

$$\binom{2^{96}}{2} 2^{32} \approx 2^{223}$$

pairs of plaintexts of this difference. A 15-round differential will have a probability of 2^{-224}.

Also, there is the following 2-round iterative differential, Ω_2 of probability 2^{-32}

$$\begin{array}{lcll} (0,b,c,d) \rightarrow (e,f,0,b) & : & (0,b) \rightarrow (0,i) \rightarrow (i,2i), & p=1 \\ (e,f,0,b) \rightarrow (0,h,e,f) & : & (e,f) \rightarrow (-m,m) \rightarrow (0,m), & p=2^{-32}. \end{array}$$

Iterated to 15 rounds also Ω_2 will have a probability of 2^{-224}. There are 2^{223} pairs of plaintexts with the desired difference. Therefore there are totally 2^{224} pairs of plaintexts for Ω_1 and Ω_2. Altogether, for any fixed key, one can expect at least one good pair for one of the differentials, Ω_1, Ω_2 for Twofish up to 15 rounds, that is, at least one pair of plaintexts which follows the expected values in the differential in each round. The first four rows of Table 1 give the probabilities of the differentials, the number of plaintexts needed to generate one good pair, and the total number of good pairs for Twofish reduced to 9, 11, 13, and 15 rounds.

The above differentials can be extended by using a 1-round differential of probability one in the first round. One can use the following differential, $\Omega_{1,0}$

$$(0,0,a,0) \rightarrow (b,0,0,0) \quad : \quad (0,0) \rightarrow (0,0) \rightarrow (0,0), \quad p=1$$

which can be concatenated with Ω_1. And the following differential, $\Omega_{2,0}$

$$(0,0,0,b) \to (0,c,0,0) \quad : \quad (0,0) \to (0,0) \to (0,0), \quad p = 1$$

can be concatenated with Ω_2. This yields $(2r+2)$-round differentials of probabilities $2^{-(32r)}$. The "price" to pay for this improvement in probability (or one extra round) is fewer pairs of plaintexts with desired difference. There are $2^{63}2^{96} = 2^{159}$ pairs of plaintexts for each of the two differentials with the desired input differences, a total of 2^{160} pairs. Thus for any fixed key, one can expect to get at least one good pair for Twofish reduced to 12 or fewer rounds. The middle three rows of Table 1 give the probability of the differentials, the number of plaintexts to generate one good pair, and the total number of expected good pairs. In an attempt to get access to more pairs of plaintexts to be used in the analysis we introduce some more differentials. The following differential, $\Omega_{1,1}$

# Rounds	Proba- bility	# Plaintexts to generate 1 good pair	Total no. good pairs	Differentials
9	2^{-128}	2^{64}	2^{96}	Ω_1^* and Ω_2^*
11	2^{-160}	2^{80}	2^{64}	
13	2^{-192}	2^{96}	2^{32}	
15	2^{-224}	2^{128}	1	
8	2^{-96}	2^{64}	2^{64}	$\Omega_{1,0} \mid \Omega_1^*$, and $\Omega_{2,0} \mid \Omega_2^*$
10	2^{-128}	2^{96}	2^{32}	
12	2^{-160}	2^{128}	1	
8	2^{-128}	2^{64}	2^{128}	$\Omega_{1,1} \mid \Omega_1^*$, and $\Omega_{2,1} \mid \Omega_2^*$
10	2^{-160}	2^{80}	2^{96}	
12	2^{-192}	2^{96}	2^{64}	
14	2^{-224}	2^{112}	2^{32}	
16	2^{-256}	2^{128}	1	

Table 1: Truncated differentials for different number of rounds of Twofish. The second column is the probability for each one of two differentials, the third column the number of plaintexts required to generate one good pair, and the fourth column the expected total number of good pairs. Ω^* means Ω concatenated with itself some number of times, '|' means concatenation of differentials.

$$(u,v,w,x) \to (a,0,u,v) \quad : \quad (u,v) \to (u',v') \to (y,z), \quad p = 2^{-32}$$

can be concatenated with Ω_1 and the following differential, $\Omega_{2,1}$

$$(u,v,w,x) \to (0,b,u,v) \quad : \quad (u,v) \to (u',v') \to (y,z), \quad p = 2^{-32}$$

can be concatenated with Ω_2. In both cases one gets $2r$-round differentials of probability 2^{-32r}. The advantage of this approach is that more pairs of plaintexts can be used. There are approximately 2^{255} pairs of plaintexts with the desired difference.

The last five rows of Table 1 give the probabilities, the number of plaintexts to get one good pair, and the expected number of total good pairs. As seen, for Twofish with the full 16 rounds, for any fixed key, one can expect to get one right pair following one of the two differentials. This does not necessarily mean that such good pairs can be exploited in a cryptanalytic attack. However it is surprising, in our opinion, that it is possible to push non-trivial information through all 16 rounds of Twofish.

5 Distinguishing Twofish from a random permutation

In this section we use the results of the previous section to try and distinguish Twofish from a randomly chosen permutation.

In a previous version of this note [4] we outline an attack using the differentials of the prevous section to distinguish Twofish reduced to 9 or fewer rounds. This attack considered only the zero-valued words in the differential, an approach which turned out not to work, which was explained in a rump-session talk at AES3[5]. In the same talk we argued that distinguishing attacks might still work by incorporating the nonuniform distribution of differences in the nonzero words of the differential from the previous section, e.g., the values e and f from above.

Consider the differential Ω_1, restated here for convenience.

$$\begin{array}{lcll} (a,0,c,d) \to (e,f,a,0) & : & (a,0) \to (i,0) \to (i,i), & p = 1 \\ (e,f,a,0) \to (h,0,e,f) & : & (e,f) \to (2m,-m) \to (m,0), & p = 2^{-32} \end{array}$$

As mentioned earlier the values of e and f cannot be determined directly from the values of c, d, and i. This is because the differences considered here are defined by subtraction modulo 2^{32}, but plaintext halves are combined with the exclusive-or operation. (In addition there is a one-bit rotation affecting the value of e.)

However, the values of e and f are not random for given c, d, and i. To illustrate this, let us consider a modified variant of Twofish. Instead of combining the plaintext halves via the exclusive-or operation assume that the halves are added (word-wise) modulo 2^{32}. Also, we shall ignore the one-bit rotations. Note that this yields a valid block cipher. In this case the 2-round differential will be

$$\begin{array}{ccll} (a,0,c,d) & \to & (c+i,d+i,a,0) & p = 1 \\ (c+i,d+i,a,0) & \to & (a+m,0,c+i,d+i) & p = 2^{-32}, \end{array}$$

where we have omitted to specify the differences inside the round function. In this case the differential predicts not only 32 bits of information in the round function of each round, but totally 64 bits in the ciphertexts. Note that $c - d$ can be assumed to be known in a known or chosen plaintext attack. This property iterates to any number of rounds.

It is well-known that the exclusive-or operation and addition modulo 2^{32} are closely related with respect to differentials, see e.g. [10, 8], therefore the values of e and f will depend on the values of c, d and i.

5.1 Tf-32 - a scaled-down variant

In an attempt to estimate the nonuniformness of the values in the above differentials we try to make a scaled-down version of Twofish. We shall construct a variant using 8-bit words instead of 32-bit words. However, for such a scaled-down version to have exactly the same structure as Twofish would mean that the 8-bit permutations in the round function be constructed from 1-bit permutations. Clearly such a variant is weak and it seems difficult to make a realistic scaled-down version of Twofish to 32-bit blocks.

Instead we shall choose the 8-bit permutation, called $\mathbf{g}$ above for Twofish, at random. The second 8-bit permutation, called $\mathbf{g_8}$ for Twofish above, is constructed from the first one by rotating the inputs 2 positions to the left. All additions modulo 2^{32} in Twofish are replaced by additions modulo 2^8. This also defines the PHT-tranformation. Let us denote such a scaled-down variant by Tf-32.

Note that Tf-32 does not resemble real Twofish, since for the latter the functions $\mathbf{g}$ and $\mathbf{g_8}$ are not randomly chosen 32-bit permutations. In fact, cf. earlier, the functions are built from 4-bit permutations and are far from "random".

We conjecture that if there is an attack which can distinguish Tf-32 from a randomly chosen 32-bit permutation, then there is also an attack which can distinguish Twofish from a randomly chosen 128-bit permutation. This is because one would expect, we think, that a Twofish variant with truly random 32-bit permutations will be stronger than Twofish. And for similar reasons, if a distinguishing attack on Tf-32, the scaled-down version of Twofish, does not work, one can**not** transfer the conclusion to Twofish.

5.2 χ^2-tests

To support our claim from [5] that distinguishing attacks on Twofish based on the differentials of the previous section might work, we implemented tests on versions of Tf-32.

We used the following 4-round differential, where we have used similar notation as earlier in this note:

$$(0, 0, a, 0) \to (b, 0, 0, 0) \to (c, d, b, 0) \to (e, 0, c, d).$$

For real Twofish this differential has probability 2^{-32}, thus it predicts that 32 particular bits of a pair of ciphertexts will be equal, thus similar as for a randomly chosen permutation. For the reduced variant the probability is 2^{-8}. We argue (as in [5]) that the distribution of the value e, c, d (a three-byte value) is nonuniform and that this can be used to distinguish this variant from a randomly chosen permutation. The attack goes as follows. Choose pairs of plaintexts which differ only in the third word (byte). If the pair of ciphertexts has equal values in the second word, record the values of the difference in the first, third and fourth words (e, c, d above).

One possible tool to measure nonuniformity is the χ^2-test [12, 9]. In a χ^2 test, the observed χ^2 statistic is compared to $\chi^2_{a,m-1}$, the threshold for the χ^2 test with $m - 1$ degrees of freedom and with significance level a. For the Twofish variant we choose a χ^2-test with $2^{24} - 1$ degrees of freedom. For a randomly chosen function one would get a χ^2-value of $2^{24} - 1 = 16, 777, 215$ in 50% of the cases. We give here other significance levels for tests with $2^{24} - 1$ degrees of freedom.

Level	0.50	0.60	0.75	0.90	0.99
χ^2	16,777,215	16,778,682	16,781,122	16,784,639	16,790,694

This means that for a randomly chosen function, a χ^2-value of 16,790,694 will only happen in 1% of all cases.

Now back to the test on the Twofish variant.

5.3 The attack

For Tf-32 we choose a structure of 2^8 plaintexts all with equal values in the first, second and fourth bytes, and with different values in the third bytes. From such a structure we can form $2^{15} - 2^7$ pairs with the above input difference to the truncated differential. For each pair of ciphertext we record the exor-difference of the values of the first, third, and fourth bytes, if and only if, the value of the second byte is zero. If we assume that the second byte of a ciphertext difference is zero with a probability of 2^{-8}, which was confirmed in the tests below, each structure will give 2^7 pairs for the χ^2 test. With 2^7 structures, i.e., 2^{15} texts, one gets about 2^{14} pairs for the χ^2 test. Here we give the χ^2 values using 2^{15}, 2^{16}, 2^{17}, and 2^{18} texts, respectively. The values are averages over 10 different tests, where in each test we chose a random 8-bit permutation to be used in the round function and random round keys.

# plaintexts	2^{15}	2^{16}	2^{17}	2^{18}
χ^2	16,777,821	16,785,130	16,788,487	16,796,900

In the test with 2^{16} texts, in nine of ten tests the χ^2-value was well above 16,777,215. The above tests clearly demonstrate that the values of the differences of the differentials for four rounds of encryption are very nonuniformly distributed.

For Twofish one would choose structures of 2^{32} plaintexts from which one can construct $2^{63} - 2^{31}$ pairs of the desired difference and proceed in similar way as above.

5.4 Variants

It is difficult to conduct similar tests on (real) Twofish, since an analogue implementation would require a table of 2^{96} entries. However there are possible variants of the attack. As an example, we implemented distinguishing tests on Tf-32 where we recorded only the values of the third and fourth words in the differentials (the values c and d above). In this case we use a χ^2 test with $2^{16} - 1 = 65,536$ degrees of freedom. In 10 such tests with 2^{15} plaintexts we obtained an average χ^2-value of 65,571.

There are many other possible ways to set up such attacks. For example, one could look at only a subset of bits from all three output bytes, and/or incorporate also the value of the nonzero plaintext word/byte of the plaintext difference ("a" above).

5.5 Conclusion of the attack

As mentioned we think it is very likely that the above attacks will work also for real Twofish, at least when reduced to 4 rounds. For more than 4 rounds, it is an open question of how nonuniform the distribution of differences will be. Unfortunately, it seems there is not much hope that we can answer these questions, since it is difficult to conduct such tests even for 4 rounds. The points we wanted to make are, first, that the distribution of the differences in the differentials detected for Twofish in this note are indeed nonuniform, and second, that the "retreat" from the Twofish team [3] that distinguishing attacks on 4 rounds of a Tf-32 version show no deviation is wrong. In fact it has been clearly demonstrated that there are attacks showing a nonrandom behaviour. The reason for the pessimistic result of [3] is that it did not incorporate the nonuniformness of the nonzero values in the used differentials.

One additional conclusion is that although the mixed use of group operations might help to make it difficult to mount attacks on an algorithm, the work in this paper also shows that it makes it difficult to conclude resistance against attacks.

6 Open Problems

In the differentials of this paper and in the above distinguishing attacks we have not taken advantage of any intrinsic properties of $\mathbf{g}$ and $\mathbf{g}_8$. However, as shown in Sections 2 and 3 there are properties of the Twofish round permutation which are not present in a randomly chosen permutation. We are convinced that the attacks as described above would work for (real) Twofish despite of these facts, and also that incorporating intrinsic properties of $\mathbf{g}$ and $\mathbf{g}_8$ will only improve on our results, not the opposite. Also, we note that the designers themselves[16, §8.3.1] have found some interesting high-probability truncated differentials through the whole round function, called F in [16].

7 Conclusion

In this paper we analysed the AES-candidate Twofish. First, we showed that the Twofish S-boxes have properties not present in randomly chosen S-boxes, and that the linear transformation allows for truncated differentials of probability one. Second, we showed that there exists differentials for Twofish for up to 16 rounds, predicting at least 32 bits of nontrivial information in every round. Moreover, the probabilities of these differentials are high enough such that one can expect to find one good pair of plaintexts following the differential through all 16 rounds for any fixed key. This is mainly due to the structure of the round function of Twofish. In this part of our analysis we did not make use of any intrinsic properties of the S-boxes and linear transformations in Twofish. We believe that our findings will only be improved taking such an approach. Third, we showed that it is possible to use the differentials to distinguish a scaled-down version of Twofish reduced to four rounds from a randomly chosen permutation contrary to what has been claimed by the designers. We are convinced that such attacks would apply to (real) Twofish as well. It is left as a (presumably hard) open problem to determine for how many rounds the distinguishing attack will work.

Acknowledgments

The author would like to thank Eli Biham, Don Coppersmith, Willi Meier, Håvard Raddum, and Vincent Rijmen for helpful comments.

References

[1] J. Daemen, L. Knudsen, and V. Rijmen. The block cipher Square. In E. Biham, editor, *Fast Software Encryption, Fourth International Workshop, Haifa, Israel, January 1997, LNCS 1267*, pages 149–165. Springer Verlag, 1997.

[2] J. Daemen and V .Rijmen. AES proposal: Rijndael. Submitted as an AES Candidate Algorithm. Description available from NIST, see `http://www.nist.gov/aes`.

[3] The Twofish Team. A Second Twofish Retreat. Presented by Niels Ferguson at rump session of AES3. `http://csrc.nist.gov/encryption/aes/round2/conf3/aes3agenda.html`

[4] L.R. Knudsen. Trawling Twofish. Technical report # 189, April 2000. University of Bergen, Department of Informatics.

[5] L.R. Knudsen. Trawling Twofish (revisited). Presentation at rump session of AES3. `http://csrc.nist.gov/encryption/aes/round2/conf3/aes3agenda.html`

[6] L.R. Knudsen. New potentially weak keys for DES and LOKI. In A. .De Santis, editor, *Advances in Cryptology: EUROCRYPT'94, LNCS 950*, pages 419–424. Springer Verlag, 1995.

[7] L.R. Knudsen and T. Berson. Truncated differentials of SAFER. In Gollmann D., editor, *Fast Software Encryption, Third International Workshop, Cambridge, UK, February 1996, LNCS 1039*, pages 15–26. Springer Verlag, 1995.

[8] L.R. Knudsen and W. Meier. Improved differential attack on RC5. In Neal Koblitz, editor, *Advances in Cryptology - CRYPTO'96, LNCS 1109*, pages 216–228. Springer Verlag, 1996.

[9] L.R. Knudsen, and W. Meier. Correlations in RC6. To appear in proceedings from Springer Verlag of Fast Software Encryption Workshop No. 7 held in New York, April, 2000.

[10] L.R. Knudsen, V. Rijmen, R.L. Rivest, and M.P.J. Robshaw. On the design and security of RC2. In S. Vaudenay, editor, *Fast Software Encryption, Fift International Workshop, FSE'98, Paris, France, March 1998, LNCS 1372*, pages 206–221. Springer Verlag, 1998.

[11] L.R. Knudsen, M.P.J. Robshaw, and D. Wagner. Truncated differentials and Skipjack. In M. Wiener, editor, *Advances in Cryptology: CRYPTO'99, LNCS 1666*, pages 165–180. Springer Verlag, 1999.

[12] D.E. Knuth. The Art of Computer Programming, Vol. 2. Addison-Wesley, 1981.

[13] J.L. Massey. SAFER K-64: A byte-oriented block-ciphering algorithm. In R. Anderson, editor, *Fast Software Encryption - Proc. Cambridge Security Workshop, Cambridge, U.K., LNCS 809*, pages 1–17. Springer Verlag, 1994.

[14] R. Merkle. Fast software encryption functions. In A.J. Menezes and S.A. Vanstone, editors, *Advances in Cryptology - CRYPTO'90, LNCS 537*, pages 476–501. Springer Verlag, 1991.

[15] National Institute of Standards and Technology. Advanced encryption algorithm (AES) development effort. `http://www.nist.gov/aes`.

[16] Schneier, Kelsey, Whiting, Wagner, Hall, and Ferguson. Twofish: A 128-bit block cipher. Submitted as candidate for AES Available at `http://www.counterpane.com/twofish.html`.

[17] Schneier, Kelsey, Whiting, Wagner, and Ferguson. Comments on Twofish as an AES candidate. Document, March 24, 2000. To be presented at AES3, April 2000.